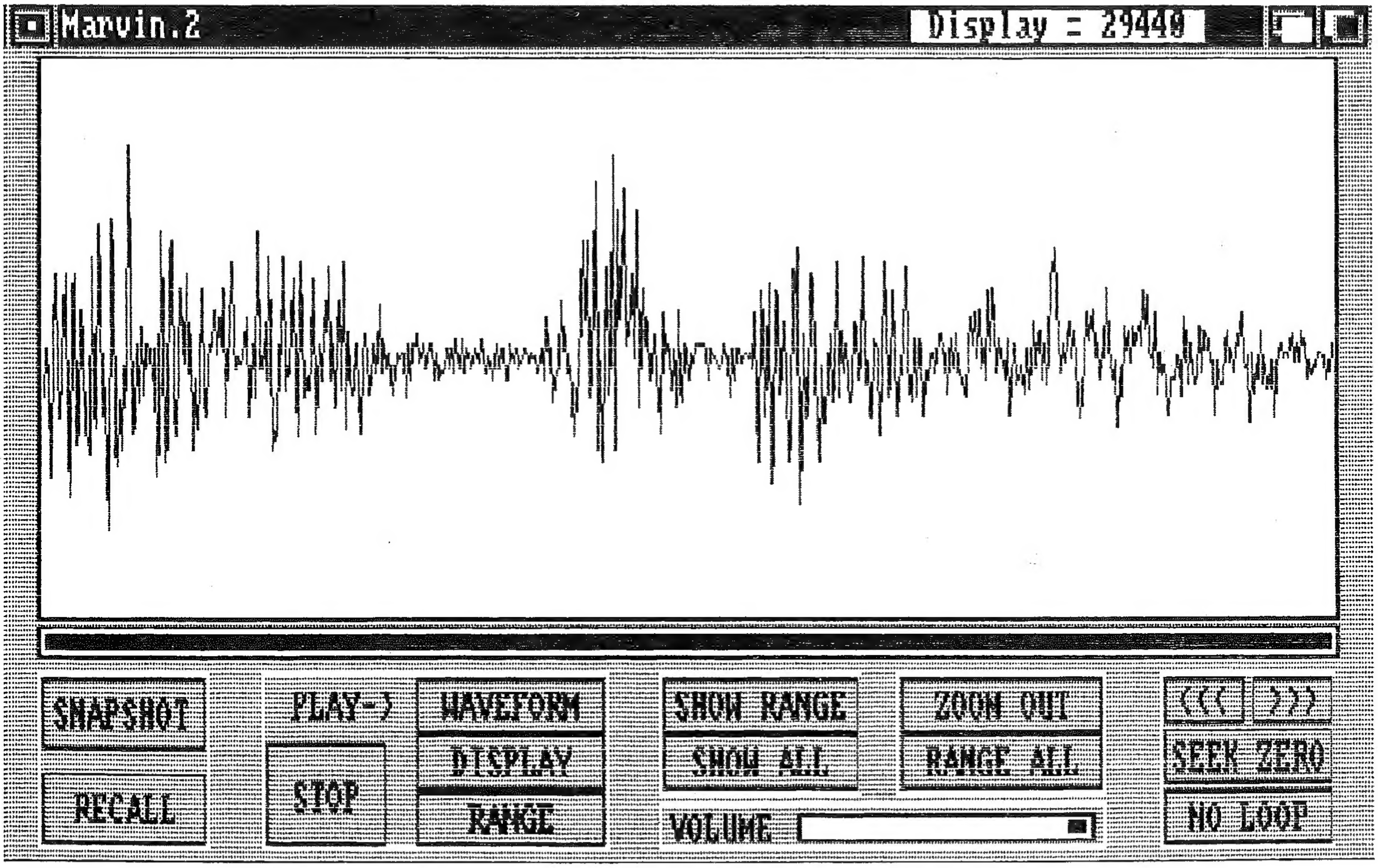
AMIGA

WORKBENCH

$2
RRP

Registered by Australia Post — Publication No. VBG7930

Number 25 Circulation: 1100 June 1988



Next AUG Meeting

Sunday, June 19th, 1988 at 2pm

(Doors open at 1pm, meeting starts at 2pm sharp)

AUG meetings are held in the Rotunda at Monash University
Wellington Road, Clayton Melways map 70 reference F10 and map 84A

Amiga Users Group Inc, PO Box 48, Boronia, 3155, Victoria, Australia

Australia's Largest Independent Association of Amiga Owners
The *Amiga Users Group Inc* has no affiliation with Commodore

Who Are We?

The Amiga Users Group is a non-profit association of people interested in the Amiga computer and related topics. With over 950 members, we are the largest independent association of Amiga users in Australia.

Club Meetings

Club meetings are held at 2pm on the third Sunday of each month in the Rotunda at Monash University, Wellington Road, Clayton. Details on how to get there are on the back cover of this newsletter. The dates of upcoming meetings are:

Sunday, June 19th at 2pm
AGM AGM AGM **Sunday, July 17th at 2pm** AGM AGM AGM
Sunday, August 21st at 2pm

Production Credits

This month's newsletter was edited by Peter Jetson. Equipment and software used was: TurboDOS S-100 computer, Brother HR-40 printer, Gemini 10x printer, Wordstar, Fancy Font and Grabbit.

Copyright and Reprint Privileges

Contributions

Articles, papers, letters, drawings and cartoons are actively sought for publication in Amiga Workbench. Please submit your contributions on disk, since that means they don't have to be re-typed! All disks will be returned! Please save your article in **text-only** format (If it can be loaded by ED, it is text-only). Absolute deadline for articles is 16 days before the meeting date. Contributions can be sent to: The Editor, AUG, PO Box 48, Boronia, 3155.

Membership and Subscriptions

Membership of the Amiga Users Group is available for an annual fee of $20. To become a member of AUG, fill in the membership form in this issue (or a photocopy of it), and send it with a cheque for $20 to:

Amiga Users Group, PO Box 48, Boronia, 3155

Public Domain Software

Disks from our public domain library are available on quality 3.5" disks for $8 each including postage on AUG supplied disks, or $2 each on your own disks. The group currently holds over 180 volumes, mostly sourced from the USA, with more on the way each month. Details of latest releases are printed in this newsletter, and a catalog disk is available.

Member's Discounts

The **Amiga Users Group** negotiates discounts for its members on hardware, software and books.

Currently, **Technical Books** in Swanston Street in the city offers **AUG** members a 10% discount on computer related books, as does **McGills** in Elizabeth Street. Just show your membership card. Although we have no formal arrangements with other companies yet, most seem willing to offer a discount to AUG members. It always pays to ask!

Back Issues of Newsletter

All back issues of Amiga Workbench are now available, for $2 each including postage. Note that there may be delays while issues are reprinted. Back Issues are also available at meetings.

AmigaLink - Our Bulletin Board System

The Amiga Users Group operates a bulletin board system devoted to the Amiga, using the Opus message and conferencing system. AmigaLink is available 24 hours a day on (03) 792 3918, and can be accessed at V21 (300bps), V22 (1200bps), V23 (1200/75bps) or V22bis (2400bps) using 8 data bits, 1 stop bit and no parity.

AmigaLink is part of the world-wide Fido/Opus network of bulletin boards, and we participate in the national and international Amiga conferences. AmigaLink has selected Public Domain software available for downloading, and encourages the uploading of useful public domain programs from its users. AmigaLink is FidoNet node number 631/324.

Newsletter Advertising

The Amiga Users Group accepts commercial advertising in Amiga Workbench subject to the availability of space at these rates:

Quarter page	$20
Half page	$40
Full page	$70
Double page spread	$120

These rates are for full-size camera-ready copy **only**. We have no photographic or typesetting facilities. Absolute deadline for copy is 16 days before the meeting date. Send the copy and your cheque to: The Editor, AUG, PO Box 48, Boronia, 3155, Victoria.

Amiga Users Group Committee

Co-ordinator:	Bob Scarfe	376 4143	Kensington
Vice Co-ord:	Fergus Bailey	211 7845	Malvern
Meeting Chair:	Ron Wail		
Secretary:	John Elston	375 4142	M' Ponds
Treasurer:	(temporarily vacant)		
Membership:	Neil Murray	794 5683	Dandenong
Purchasing:	Bohdan Ferens	792 1138	Dandenong
Book Library:	Joan Wood	580 7463	Aspendale
Disk Library:	Geoff Shiell	578 8362	Brighton
	Margaret Bedson	578 8362	Brighton
Editor:	Peter Jetson	762 1386	Boronia

Arkanoid (or Breakout Grows Up)

by Stan Thomas

The very first computer games consisted of variations on the idea of using one or two small white bats to bounce a small white ball around the screen. These were followed by, among other things, a game called Breakout, which took the idea further by adding a collection of bricks that had to be eliminated by hitting them with the same small white ball, which you had to keep in play as it bounced around the screen.

Now, after years of neglect while computer games increased in variety, complexity and sophistication, the Breakout idea has resurfaced in brilliant style in the form of Arkanoid. The basic idea is still the same, in that you have to eliminate a screen full of bricks. The difference now is that a whole swag of new elements have been added to the gameplay. Some of the bricks you hit have special properties that can change the way your bat (a space pod called a Vaus in the game's scenario, but who cares?) or ball behaves, such as slowing down the ball, widening the bat, turning it into a laser so that you can shoot the bricks down, or splitting the ball into three. There are obstacles that float around the screen, generally trying to get in the way and bouncing your ball off in random directions.

I haven't seen the arcade version of this game, but I believe that this version matches it in quality. And what quality! This is the first game I've seen on the Amiga that uses overscan. The play area takes up the entire screen, making the screens used by other games seem small by comparison. The graphics are superb, and make the best use of drop shadows I've ever seen. Even the little white ball has its own drop shadow!

The only way to control the bat is via the mouse, but this is such an easy and natural operation that this is no limitation at all. This is easily the best, most addictive game I've ever seen, for the Amiga or any other computer. And it is simple enough at its simplest level for me to get through all 32 screens and beat the final screen which crowns the game.

Another marvellous feature is that you can start each game from where you left off. The one flaw with Marble Madness is that you have to start every game from screen one, which I find very annoying. That the writers of Arkanoid didn't do this is a big big plus in their favour.

Ten out of ten for this superb, must-buy game. Arkanoid is published by Discovery Software International, and is distributed in this country by Ozisoft. Make sure you get the PAL version, since I believe the American version might be floating around in some places.

==

Two Years After

by Detlef Pelz

I've had my A1000 for over two years now and I've been a member of the AUG since its inception. In that time I've been to only one meeting - I'd come to more meetings but Monash is a bit far to travel. It would be more of an incentive if I had some idea of what each month's agenda was, so that I could make some effort to attend whenever there are items of interest to me on the agenda.

In the time that I've had my Amiga I've used it mainly to "play" with; "play" in this context includes learning to program (albeit only in Basic), playing games, and communicating with others through Viatel and BBSes. So far I've managed to "talk" to people in most Australian cities, and to people in the US and Canada, principally Toronto, where my wife hails from.

I don't know how I'd classify myself on the scale of Amiga owners - I'm not a hacker by any means, nor am I terribly interested in many of the fancy things the Amiga can be made to do, usually incurring costs for additional hardware and software that add up to more than my basic machine plus an extra disk drive cost in the first place. When I think about it, I probably would've been happy with a PC AT or equivalent, since that's one of the machines I use in my job as a computer systems manager cum programmer at a Melbourne typesetting house. So why the Amiga?

I was initially attracted by the rave reviews in local and overseas mags extolling the virtues of what is, after all, one of the finest machines of its type around. And of course the price was right. But it's not really something I had to have for any other reason than that I liked the specs and I liked the price. The only other computer I've ever owned was an Acorn BBC-B, which had a massive 32k RAM and one of the nicest BASICs I've ever encountered on any machine I've ever used, and the list grows constantly.

Moving soon?

Don't forget to tell us!

Every month, Australia Post returns newsletters to us marked "Left Address", "Not At This Address" or "Return to Sender".

To make sure this doesn't happen to your newsletter, please tell us if you move!

If possible, include a mailing label from a past newsletter or your membership number.

Some things about the Amiga do disappoint. For instance, the cost of peripherals such as hard disks - almost a "must" these days - causes bouts of apoplexy in all but the financially foolhardy or millionaires. Often they are one and the same - maybe that's why they are millionaires. But I do blanch when the cost of an add-on exceeds the limit of my bankcard. Will I be in my dotage before a 20Mb drive for the Amiga costs as little as one for a PC?

Another thing that makes me pine is the lack of a really good PC compatibility. I'd like to be able to take some of my stuff from work and "play" with it at home, but the Sidecar is perhaps better suited to a Honda while the Bridgeboard is somewhat less than hugely compatible. One of the main problems seems to be the screen updating being woefully slow. And those floppies! Why are they so slow? I believe the 1.3 revision of AmigaDos will address the problem of slow disk IO; I certainly hope so, I do expect something a bit better from a machine of this calibre. [Editor's note: AmigaDOS V1.3 **will not** speed up floppy disk IO, but **will** speed up hard disks about seven times.]

Apart from these grumbles, I wish things like extra memory were a bit easier on the wallet. Memory is as important as a hard disk for most users and really shouldn't cost any more for us than it costs other brands' users. It wouldn't be a problem if the Amiga were sprung from the same loins as PCs - but then it wouldn't be an Amiga, would it? One way that Commodore could have overcome the problem in the first place would have been to include a text-only mode in the menu, a-la-BBC, leaving the more memory-hungry screen modes for those programs that need them. I think that I am not the only one who writes useful little programs in BASIC that don't have any need of fancy graphics requiring heaps of memory. As it is, I find it astonishing that Commodore still insist on marketing the machine with a paltry 512k as standard. Even PCs come better equipped than that!

And what of the future? I don't think I'll bother spending heaps on extras for my A1000. I'm still using the same old trusty Epson RX-80 I bought back in 1982 for my BBC, and my modem will work with future machines, should I buy one. What I'm really waiting for is for Commodore to get it right with their next Amiga, the 2500 or 3000. It should have a minimum 1Mb RAM, fast floppies, and preferably a 30 or 40Mb hard disk as standard; AmigaDos MUST eventually be seen up there with MS-DOS and Unix as a preferred operating environment, keeping to the present scheme of catering to both mouse and finger users. The PC compatibility issue must be resolved with the introduction of true and transparent compatibility - perhaps then we'll start seeing PCs with Amiga compatibility as an option.

Until the above eventuates, I'll keep "playing" with my Amiga just as it is, and I might even buy a cheap PC AT clone as my "other" machine unless Commodore comes to the party in the next 6-12 months. For the nonce I couldn't really recommend the Amigas to anyone but ardent gamesters or those inclined to explore the more esoteric possibilities the machine presents. As for business use, I opted for a couple of locally assembled PC AT clones. I guess that speaks for itself.

==

The North West Amiga Users Group

by Simon Shead

This article is written for those of you who are interested in the N.W.A.U.G. meetings, as well as those of you that have had questions about the group's activities, but were afraid to ask. The aims and aspirations of the group are better dealt with in a separate article, so I won't go into that here. Rather, this is a "diary" of events in the N.W.A.U.G. calendar.

The inaugral meeting of Amiga users in the North-Western suburbs was announced by Neil Beatty, and as a result, approximately 35 people turned up at a small room in the Essendon Community Centre on the 19th February, 1988. Matters discussed at this meeting included - whether we wanted to be part of the Amiga Users Group or conduct ourselves as an independent body, finances, our aims, nomination of office bearers, and that elusive one - a name for the group. After much debate, 'North West Amiga Users Group' was agreed upon. In all, this meeting was quite successful in organising a structure and direction for the group.

The next meeting, on the 2nd March, was held in the same 'dungeon'. On this day, we were shown a demonstration of "Pagesetter" by Hugh Leslie, starting from scratch and building a NWAUG newsletter while we all looked on. Most people were amazed how easy it was to create a newsletter in just 15 minutes. George Wahr demonstrated his PROTON memory expansion and explained the advantages of extended memory and RAM: disks. As well, there were many demonstrations in the background, and people talked to other members about hints and problems. Although the lack of a phone line prevented us from having a modem display on this night, it was very successful.

On the 16th March, in a larger room on the second floor, Bob Scarfe presented a comprehensive demonstration of available music software, how music programs have evolved over the last year and a half or so, starting with Music Studio and ending with the sequencing program CQUIN. The newly released JET flight simulator was presented on Bob Scarfe's projection TV screen by Simon Shead, the Top Gun of the NWAUG showed us all how to blast those MiG fighters, and land on the aircraft carrier.

On 30th March, Neil Beatty presented GALILEO, the astronomy program, and showed some of the finer points in stargazing. John Phipps gave a small talk on some Public Domain utilities available through the club, along with a comment on the policy used for articles to be printed in "WORKBENCH". After a somewhat lengthy delay, George Wahr graced us with his presence, with 50 copies of the first NWAUG newsletter in tow. Then we presented our "MARBLE MADNESS" competition, where members took turns to eliminate their opponent, in an elimination series. Those who played appeared to be quite excited. The NWAUG library was also officially started.

On 13th April, guest speaker Peter Jetson spoke about how he produced the WORKBENCH magazine and the editorial policy. Hugh Leslie showed us the innards of an Amiga 1000, and described the PAL chip modification and Spirit memory board installation he had just completed. Also on display was a PROTON memory expansion fitted to an Amiga 500. Bob Scarfe showed us the basics on using spreadsheets on his large projection television, and how easy it was to turn out stunning graphs on any statistical information. Bob Laidlaw then took a smaller group on the finer points of spreadsheets, while the second NWAUG competition, JET, was conducted. After tense moments and a couple of replays, S. Kernaghan was declared winner, and was presented with a certificate and a prize of five NASHUA disks.

On 27th April we had a demonstration from Hugh Leslie of PHOTON PAINT, and those present were impressed by the features of this program, as he created 3D shapes and wrapped flat pictures onto them. Simon Shead spoke on the legal aspects of copyright notices on software, copying software and the Australian law in relation to computer programs. John Elston spoke on the history and basics of computers, and the connection with the Amiga. Greg Hudson demonstrated AUTO-KICK, a modified kickstart disk for A1000 owners. He also spoke on the RAM chip shortages. Our first attempt at desktop video was presented, with the BADGE killer demos from the Fish Disks compiled onto videotape and shown on the projection TV - the members were quite impressed with the outcome.

On 11th May, we had an extended question and answer session, with many newer members asking questions that the old hands could readily answer. We announced the soon-to-be acquired 5.25 inch disk drive for our public domain library, and we showed a virus-killing kickstart. George Wahr demonstrated Transformer, and explained how IBM emulation on the Amiga was achieved. Our Shanghai competition was conducted in 20 second limit tournament-mode - after one or two ties and replays, Neil Beatty was announced winner and presented with his prize of Nashua disks and Certificate.

The next meeting dates are 25th May, 8th & 22nd June, 6th and 20th July. They are held every second Wednesday at 7.30 p.m. at the Essendon Community Centre in Mount Alexander Road, Moonee Ponds in rooms 19 & 20. Everyone is welcome to attend.

==

A short note in reply to some Amiga questions

by Donald Welsh

(This article pretends to answer a few questions. It even pretends to be meaningful ... decide for yourself.)

In Workbench issue # 23 was an article by Willie C. de Lyte about programming in Amiga Basic. In it, he asserts that the "painfully slow scrolling" in Amiga Basic is due to its keyword checking. The interpreter does all its keyword checking when you move from one line to another -- this could not have any effect on the scrolling speed. The slow scrolling is due to Amiga Basic using "smart refresh" in an attempt to save memory.

Smart refresh means that Intuition does not reserve memory for repair of windows which have been obscured and uncovered. The application, in this case Amiga Basic, keeps track of what should be in the window and refreshes it as needed. Unfortunately, this is slow. This design decision was made because the original Amigas were supplied with 256K memory, although everyone I ever knew bought the upgrade to 512K with the machine.

In a letter to the Editor, Sandy Gray asks questions many of you must have asked at one time or another. Let me summarize in case you missed it:

Where can I find listings of the library modules for Amiga Basic, descriptions of their functions and examples of how to use them? ... descriptions of the file formats for ACBM and ILBM?

Where do I find out about Intuition, windows, screens, gadgets and how to use them from within BASIC or C?

Where can I see reviews of the books in the bibliography at the back of the Amiga manuals?

Why is this information not supplied with the Amiga in the first place?

Whew! Well, Sandy, it's like this....

All the standard functions and libraries are documented in the RKM (ROM Kernel Manual). This is where you will find more than you ever wanted to know about Intuition, etc., and how to call it all from C. The IFF standard is also discussed therein, with detailed descriptions of FORM ILBM. FORM ACBM is an alternative to FORM ILBM which was developed for use with Amiga Basic. The Basic programs on the Extras disk have all the information about ACBM anyone outside C-A Technical Support would ever need. They also provide a good example of how to write a program to read and write ILBM images.

As to why all this isn't supplied with the Amiga, well, that's quite a question! In brief, it's because it would be far too intimidating to the casual or novice user. Remember, the Amiga has a mouse and windows to make things easy for people who don't want to have to read a thousand pages, or even a hundred, or even ten, before beginning to use a computer.

There are lots of users who would look at a set of such manuals arriving with the computer and say, "I won't read all that. I won't read any of that. Why was it included? Why did *I* have to pay for it?" The information on basic CLI commands was omitted from the original manuals for ordinary Amiga users.

Where can you find reviews of these books? What have you been reading? (Just kidding...!) Try looking in Byte or even ... Amiga World! Amazing Computing is good, and I've heard of other Amiga-oriented magazines floating around. While I'm on the subject ... remember that Amiga World was launched before the Amiga 1000.... The Amiga -- the computer with a magazine devoted to it before it was born!

Which books are any good? First of all, I avoid books with "Introduction", "Guide", "Beginner's", or "Elementary" in the title. If you can wade through the RKM you'll find what you were asking for before. As ordinary mortals like you and me (i.e. non-developers) can scarcely understand most of the RKM, there are other books out there which explain much of importance, some even containing *working examples*. A computer book never has too many examples. Look for books with lots of examples, preferably ones that are interesting to you or deal with areas you've had problems in.

As to specific books, ask around at the meetings and look around at the bookstores and write to Workbench about what you find. We all need more articles! Until then, I'll leave you with comments on specific books I've seen.

The Mortimore book is okay, containing good summaries of the ROM library calls from C and assembler. There was a version for 1.1, and another for 1.2. With 1.3 in the hands of the developers, due to be given to the rest of us Real Soon Now, and 1.4 looming on the horizon, I can't say whether or not you should buy a book like this now or later. (By the way, long ago, in another galaxy, in a newsletter far far away (Workbench Number 16 to be exact), didn't Commodore claim that AmigaDos version 1.3 is "a myth"? Hmm.)

The Compute Guide to the Amiga is not a book I'd buy. There isn't much to say about a book which tells you all about telecommunicating and graphics on the Amiga by describing one commercial package for each. There is a book of Compute programs -- many of them are games -- reprinted from the magazine. That might be worth getting, if you buy it with the disk ... those are long listings!

If you type them in, you might learn something about the proper naming of variables. Compute follows the objectionable practice of naming variables "i", "j", "k", and "l". That's okay for a while, but then you come across a statement and you ask ... is it "k = 1" or "k = 1"? Believe me, sometimes it isn't easy to figure out! If you take the trouble to read the programs, you may learn more about programming than by reading a "book of tricks".

Software Review: Ferrari Formula One

by Kenn Barry

[Editor's note: This review was posted to UseNet by Kenn Barry, and was passed on to me by Neil Murray.]

Ferrari Formula One is, quite simply, the best road racing simulation I've yet seen. It is a detailed simulation of an entire season of Grand Prix racing (the 1986 season), with accurate renditions of 16 Grand Prix courses, plus a test track at the Ferrari home base in Fiorano. The graphics are detailed and attractive, and provide useful feedback. Sound includes not only the sound of your own engine, but the engines of any other car you are close enough to hear, and the squeal from your tires if you break them loose in a corner.

I make an important distinction between games and real simulations. A simulation is a program that tries to accurately simulate the real-world behavior of the thing simulated. Many driving games, alas, do not qualify as simulations. The behavior of the car on the road is too unrealistic, or the screen view is an overhead view instead of an out-the-window view, etc. Ferrari Formula One is a good simulation. Your point of view is from the cockpit of your formula one Ferrari F186; you have standard formula one instruments, rear view mirrors which work, and an out-the-cockpit view. The handling of the car seems quite realistic (though I confess I've never driven a real formula 1 racer). The frame rate looks to be around 5 frames/sec, which is adequate for driving.

But this program includes far more than just the races themselves. You must maintain your car as well. Parts wear out (quickly, in this kind of driving!) and must be replaced. Choices must be made for the appropriate configuration of your car for each race. Mauro, the head of your pit crew, will recommend settings for the various systems, and so far I've found it wise to always heed his advice. The gear ratios can be changed, as can the wing angles, tire types (can be different for each wheel), and shock stiffness. The engine can be repaired or replaced, and different ROMs can be chosen to control the fuel system (for different fuel/air ratios). And these things MATTER. A worn engine or an unsuitable choice of tires can make victory impossible. Each course in the circuit is best taken on with a different configuration. Weather is also included. A wet track on a rainy day will require you to switch to rain tires, and take the corners much more cautiously.

There are three levels of difficulty available. These control the driving skill of your computer-controlled opponents, as well as the likelihood of your car suffering a mechanical failure. At the highest level of difficulty, you must also do your own shifting (the program shifts gears at the correct times for you at the easier levels).

But the race, of course, is the thing. You run each race against 7 other drivers, all controlled by the program. Before each race there are two practise sessions for familiarizing yourself with the course, two qualifiers (your pole position in the race is determined by your best lap time in the qualifiers), a warmup, and the race, itself. You may choose to skip these if you prefer, however. I usually skip the qualifiers and race from the back of the pack, since I find finessing my way through the other drivers to be the most exciting and challenging part of a race. The length of the race is settable, from a minimum of 18 kilometers up to a full-length Grand Prix, 2 hours.

And the race IS exciting! The sound and graphics create the right atmosphere, and the sense of driving is extremely good. Attention to detail is excellent. You can hear the other cars throttling back and downshifting as they approach a turn, giving you valuable feedback on when to start slowing, yourself. You can even see flame from the tailpipes of the car ahead each time he downshifts, another valuable cue. The rear view mirrors keep you informed of any car breathing down your neck, as well as letting you watch for wear on your rear tires.

Control of the car is via the mouse. The right mouse button is your gas pedal, the left is your brakes, and you steer with left/right movement of the mouse. There are keyboard controls for shifting (at the highest difficulty level) and for control of your turbocharger - higher settings give you more horsepower, but cost you on mileage and engine wear.

At your home base, Fiorano, there are some extra facilities not available at the Grand Prix tracks (which all have garages for work on your car). Fiorano has a test stand for checking your engine, and a wind tunnel where you can test various settings of the wings (by the way, if any of the Gentle Readers are surprised by wings on a car, look at a formula 1 racer - they have 'em, front and back).

Playability is excellent. I've probably put about 30 hours in behind the wheel, and can now usually win the races at the two lower levels of difficulty, but the highest level is still very challenging, and I suspect I will have to drive many more hours at that level to become a top competitor. Successful driving techniques are basically the same as they are in real life, as are penalties for sloppy driving. Skidding onto the shoulder in a turn will cost you much speed, and can lose you anywhere from a fraction of a second to a few second's time, depending on how far from the course you stray. A spinout will cost you even more time. And worst of all is a collision. A collision can put you in the hospital long enough to miss many races. When I raced the full season's events for the second time, I was struck from behind seconds after the race at Hockenheim began, and penalized with 47 days in the hospital, causing me to miss the next three races as well. You are not actually constrained to run a regular season, with all the races in order, but it adds to the excitement as you strive to achieve the best record for the year. The program accumulates the points for all the drivers over the course of the season, as well as a few other useful statistics in individual races, like best lap time.

Annual General Meeting & Elections

In accordance with the Rules of the Association, notice is hereby given that the second Annual General Meeting of the Amiga Users Group Incorporated will be held on Sunday, July 19th, 1988 at 2pm at the Rotunda, Monash University, Wellington Road, Clayton.

The main purpose of the AGM is to present to the members a financial statement of the affairs of the club, and to elect a new committee to run the club until the next AGM. All financial members are entitled to vote, in person or by proxy.

There will be ten committee positions to fill:

Co-ordinator
Vice Co-ordinator
Treasurer
Secretary
Membership Officer
Meeting Chairman
and four ordinary members

Nominations of candidates for election shall be made in writing, signed by two members of the Association and accompanied by the written consent of the candidate, and shall be delivered to the Secretary not less that seven days before the date fixed for the holding of the AGM. If insufficient nominations are received to fill all vacancies, the candidates nominated shall be deemed elected, and further nominations shall be received at the AGM.

Each member is entitled to appoint another member as his proxy by written notice given the Secretary no later than 24 hours before the time of the meeting, on the form described in the rules of the Association.

Agenda

1. Co-ordinator's Report

2. Consideration of Financial Statement pursuant to section 30 of the Associations Incorporation Act 1981.

3. Election of Office Bearers.

Well, any good review should include the negatives, as well, but I find it difficult to think of many. Closest thing to a bug I discovered was to have the frame rate slow way down when I was running practise laps a couple of times. It would slow for a few seconds, then return to normal, then do it again. But this has happened only rarely, and has yet to occur while I was in an actual race. I've had the program running all day at times, race after race and lap after lap, and have never had it guru.

It's difficult to say if this is as good as a driving simulation could get on an Amiga. My guess is "no", but I hope no one ever asks me to write a better one! It would be great if the frame rate were even faster, and there are a couple of additional features I would have liked. One would be a two-player mode, a la "Firepower" or the Amiga versions of Flight Simulator and Jet. Another would be the option of designing your own race track, or modifying your car to something outside the formula 1 rules. Also, the program doesn't seem to know about yellow flags, marring the realism slightly, and hills seem to be unknown, or at least not indicated by the graphics; same for banked tracks. Lastly, the program for some reason does not make you worry about your brakes. The other major systems which are subject to wear have this wear simulated, but your brakes are always fine, and require no maintenance.

One really neat improvement would be if someone built an automobile-style controller that could be plugged into the mouse port - something with brake and gas pedals, and a steering wheel. I think that "Ferrari Formula One" made an excellent choice of controller in using the mouse the way they did, the best choice possible using the Amiga's standard hardware, but a mouse is not a steering wheel. Any ambitious hardware types out there?

Last but not least, some basics: the program is copy-protected, using a key disk system; it can be backed up, but you must insert the original on request when you boot the copy. The program runs fine in 512K or in systems with expanded memory, but it takes over the machine - multitasking is out. I have not had the opportunity to run it on a 500 or a 2000, just a 1/2 meg and a 1 1/2 meg 1000, but I would assume it runs equally well on all three machines.

Some people don't care for driving simulations. If you're one of those, I'm incredibly flattered you've read this far. For everyone else, this game is recommended.

The 1988 Amiga Developer's Conference

[Editor's note: The following text is a collection of messages left on the UseNet Amiga conference recently, about the Amiga developers conference (DevCon) held in the USA in May 1988.]

From: Kim DeVaughn, Amdahl Corporation, Sunnyvale, CA

Below are some comments by Charles Conlow that cover the first couple of days of DevCon, that I picked up from a local BBS. Not a lot of real specifics, but some interesting hints, etc. I'd be interested in what other attendees have to offer (Leo, Marco, any C-A'ers or CBM'ers, etc). I believe that CBM will be making a set of DevCon notes (and floppies ?) available to those of us who (sniff) couldn't get away. Is that right, Lauren?

Friday, Day #1

Here are my notes, opinions and C-A's (Commodore-Amiga) announcements from the first day (Friday) of the 1988 Commodore Amiga Developer's Conference.

Because C-A did not anticipate such a large attendance, the whole shebang got off to a late start. I was in line to register at 8:15 for the 9:00 first speaker, and I noticed that people were STILL registering at 3:00 PM! Everything today was shifted back half an hour.

Gail Wellington gave a short welcome and introductory speech in the main conference room. It was equipped with two Amiga 2000's, a big projection screen TV and two BIG monitors (one was CBM's (Commodore Business Machines) own manufacture, a Bi-Synch Super-Hi_Res job). We watched the ever popular "Only Amiga Makes it Possible" Video, and then Gail introduced Dr. Henri Rubin.

Dr. Rueben, C.O.O. of Commodore International, gave a short speech about the MEANING of the Amiga series of machines; how far we have come with the multi-tasking aspect, and how little of the machine's promise has so far come to light. He mentioned CBM's past financial emergencies, and how the bottom line (= money) is always in our minds (CBM, Developers and third party manufacturers).

Dr. Rubin went on to say that although the Amiga has been received very well of late (in US as well as abroad), it should, and MUST! do better if we are to see the NEXT generation of Amiga computers. He also mentioned that although Amiga continues to improve and upgrade the quality of the OS, other computer manufacturers STILL can't seem to get a working multi-tasking system off the ground.

During Gail's closing remarks, members of the audience (a little over 300) stood when she named their country. Quite an international bunch this weekend... Israel, Austria, Turkey (CBM's fastest growing market), Canada, West Germany, United Kingdom, Denmark, Finland, Italy, Australia, France, Holland and Venezuela were all represented, as well as a goodly number from the good ole US of A (sitting next to me, Leo Schwab, "the man in the cape").

During the morning session, there were several sessions with speakers from CATS, C-A and CBM fielding questions from the audiences and making remarks and comments as they came to light...

Bill Koester (CATS (C-A Technical Support)) & RJ Mical (Amiga Software Engineer) led a group of first timers through the Exec - Intuition - ADos system.

Dave Haynie, George Robbins and Jeff Boyer (all C-A Hardware Engineers), led a discussion, followed by a Q & A about the A-500's design; it's expansion port

interface limits, power supply requirements and problems, and the documentation for the machine.

Dale Luck and Jim Mackraz (Amiga Software engineers) gave the Copper a going over. Speaking of what the Copper is SUPPOSED to do, and what developers are trying to make it do, they attempted to make clear the guidelines for proper use of one of the Amiga's most basic custom chips.

Next, Carolyn Scheppner (CATS) went through the latest of the MANY IFF specs; some accepted by CBM, some pending and others... well, maybe. John Toebes of Software Distillery presented their new IFF form, PGTB (ProGram Trace Back), which is of great interest to developers. This form, and the program stub that produces it, dumps to a file all the neat and interesting things that were resident in the system when it GURUed. This allows programmers to look at the environment when someone other than themselves has attempted to use their program. Watch for "Catch".

Also, Gary Bonham of Sparta delivered a speech about the ANIM IFF FORM, and its use by Aegis, Inc. He detailed its development as an outgrowth of his work with Space Defense-type systems at Sparta, and how many months of feedback and coding led to a simple, quick compression and playback technique. Bob "Kodiak" Burns raised the point that an ANIM file is not a true IFF FORM, but rather a LIST. Gary admitted that, and also admitted that an awful lot of work had gone into the thing, and was pretty far along in development to re-code for LIST. Kodiak's point, though, is well taken. IFF is SUPPOSED to be a CBM approved standard. FORMs are FORMs and LISTs aren't. They proposed to form a working group to throw this idea around. Keep tuned!

Marketing of Amiga products was also discussed with Product support people from CBM, both US and abroad. How to get the most coverage for the smallest amount of money seemed to be the jist of it, but I was listening in from another hall, and will have to get this part straightened out (later guys).

In the afternoon, Jeff Porter (Product Development, Commodore Technology Division), introduced us to the new CBM and C-A products. They were not in the room, and no price or release dates were announced. We also were told that many of these hardware thingies are still on the board, so think what you will...

We were not asked to sign any non-disclosure agreement, but I tread lightly here, as I VALUE my status as developer and a trusted member of the CADP (C-A Developer's Program). But I think I am safe in stating...

As per the Hanover Announcement, the A-2500.AT and A-2500.UX will bring IBM-PC.AT performance / or / Unix support to the A-2000. These are not NEW machines, but bundled software of CBM's already announced (but not yet available) AT card and Unix (AT & T Unix 5 release 3.1.).

Also announced was the 1.3 and 1.4 systems, with vague references to 1.5 (and you thought all they did was answer phones in West Chester!). There is an Enhanced Chip Set to be available SOON (New Fattest Agnes, New Denise and New Gary chips) to allow for 640 x 400 non-interlaced screens with 4 out of 64 colors. This chip set will also allow for software control of EITHER PAL or NTSC (did I hear someone say horizontal scrollable screens?).

Some new monitors were also discussed, which have the SMARTS to be able to handle all these new different video modes, as well as all current ones. And most (if not all) of this new technology is backwards compatible, meaning your existing 500's and 2000's can use the new chip set, monitors and enhanced OS (Unix).

Some other hardware goodies... a self contained 20 meg SCSI hard drive for the A-500. This would be the first hard drive from CBM in many years, since the 90x0 line in the late seventies. It would have a smart external power supply, that will turn itself off and on as the A-500 is turned off and on.

For the 2000, look for 2 different GenLocks (of quite different quality), the 68020 Bridge, the 80286 Bridge, the 2090A (auto-boot hard disk controller), and KickStart 1.3 in ROM (also for 500).

Yes friends, 1.3 is STILL not ready! Gamma 7 releases were made to developers, but there is still no firm date for public release for a rock solid KickStart or WorkBench/ADos for 1.3.

Late Friday, Day #1 & Saturday Day #2

Friday, in the afternoon sessions, some more technical material was discussed, including the following:

Jim Mackraz and Hedley Davis offered their (very considered) opinions of how the Amiga systems should handle FONTS / TEXT and GRAPHICS when using the newest view modes (VERY_HIGH resolution). Using the Gray Scale monitors, OverScan and HAM, system parameters may seem tossed to the wind... there is so far a general rule about this, and 1.4 should bring a STANDARD (i.e. iff or IFF.2) about FONTs, COLOR_FONTs, and VIEW_Modes.

Rob Peck (Author of the Audio Tools Library) and Dan Baker (CATS) covered the Amiga's multi-tasking limits with regard to the sounds / music she can make (and what beautiful music!). Rob Peck has been a major contributor to the Amiga Developer's world of audio, and has also followed up on his original implementation of the Audio Tools Library.

Andy Finkel has given us a pretty good idea of what 1.3 can and cannot do... There are reasons why we CAN do some things, and reasons why we CAN'T do others. Major revisions of 1.3 are the printer.device and associated drivers (quite fast, and really VERY clean for dot matrix), the FFS (Fast Filing System) for hard drive users (FFS will be standard on all devices as of 1.4) and smaller - faster C: commands (bye-bye BCPL).

Other Friday Topics were Math Libraries (including the support for the various co-processors), Creating your own libraries (remember the interrupt?), 'C' on the Amiga, What hardware implementation of 1.3 and 1.4 require to auto-boot and what 1.3 preferences is all about.

Friday, several disk of source code were distributed which contained the examples (object and source) of most of the Conference's projects, as well as Gamma 7 of WorkBench 1.3 (Still, no release date kiddies!).

Saturday, I will have to rely on other people's notes, as I had to duck in and out all day...

During the morning hours (yawns!), hardware and software was shown in the computer rom from companies such as... Aegis, MicroSmiths, Manx, Lattice, Software Distillery, and of course CBM / C-A / Commodore International.

Then Dave Haynie, George Robbins and Jeff Boyer talked about the slots of the A-2000. There MAY be mods to future machines, there probably will not, but there will CERTAINLY be a backward compatible path on this matter as the 2000 is THE AMIGA as far as C-A is concerned. The stress of this matter is that the 2000 is the ANCHOR of the series, and all future machines MUST allow for its existence.

Also on Saturday, the implementation of multiple ports was discussed. In 1.x (4, 5, 6?) there will be allowed a multitude of ports available for the end user, with a total of number of 65536. Be these serial, parallel or user, the protocol of manufacturer ID, placement, power drain and access methods all need to be set in stone... an important step toward true MULTI-USER status, especially if a windowed UNIX is to be supported.

In addition to the multiple ports issue, the idea of the TRANSPUTER was raised. Now the Amiga, already a multi-tasking machine, lends itself well to the multi-processing stage of the transputer (I know this, but you know how to :whatever: it, here it is, send it back [fixed]; thanks so much)... Well, here we go again... !

Gosh - this reporting bit is harder than I thought!

Well guys, I will finish Saturday's report tomorrow, and also the Sunday "chats" with Hardware, Software and Marketing people, as time and typing skills allow....

From: Bob Page, University of Lowell, Computer Science Dept.

It seems to me that Mr Conlow (above) read the program and reported that. Some things just didn't happen the way he reported them. For example, Rob Peck wasn't there, and Dale wasn't part of the Copper talk, just JimM. Some of my impressions, off the cuff:

* The conference notes were about 3 inches thick, single sided. Loads of useful information.
* Everyone who was at Monterey (sp?) said this DevCon was a WHOLE LOT better.
* Randy Spencer's machine taped Yar's death while we were all out partying with Discovery Software, then it was shown a few times (at least 2.5) on the big screen.
* Dave Haynie showed up with a 68030 board for the A2000, and somebody ran the C64 emulator on it when he wasn't looking. Called it the 64030. :-)
* All companies there got a KS1.3 ROM.
* Lots of companies were looking for Amiga programmers.
* Leo ran an Anim that said "OS/2 ==> HALF-OS" ... CBM let it run for a while :-) After that, Leo ran his Stars program whevener somebody let him near a machine.
* Dave Haynie won the Usenet BOING award.
* Fred Fish won an award from CBM. It was the first ever publicly presented "fatter Agnus". Six other companies got them too (picked by lottery).
* RJ Mical is expecting his second child.
* Tom Rockiki tried to describe Breshnev's line draw algorithm. Having failed that, he showed us his terrific error checking routine in BlitLab 1.3. :-)
* Mike (Powered by Cyberpunk M&Ms in training) Smithwick had Leo read the beta version of his new StarChip EnterBoing adventure at the USENET meeting. Dale kneeling before Leo?
* At the USENET meeting: CBMers Lauren, DaveB, Dale, Bryce, George (grr), Dave (boinger) Haynie, SteveB (?) and maybe more (I didn't get a lot of sleep this past weekend, kids).
* Perry & Eric advanced the concept of Amiga Working Groups, sort of like technical steering committees to hammer out ideas. Just about everybody liked the idea, including CBM.
* Randell Jesup was hired by CBM.
* Nobody wanted to talk much about IPC, multiple ports or the ultimate user interface. Well, everybody wanted to, but didn't want to get into flame wars, ala USENET. Joanne Dow kept saying "Post it on BIX! Post it on BIX!"
* Chuck McManis had a great Intuition programmer's library; showed you how to do everything. Commercial product.
* During the "beyond 1.4" talk, about 6 presenters brought up ARexx.
* Five disks were handed out: WB 1.3 gamma 7, WB Extras gamma 7, DevCon 1, 2 & 3 examples source disks. Hedley also gave out a disk on the A2024 monitor if you went to his talk.
* The SCA virus was in the computer room, on at least 6 A500s.
* All the Fish and Amicus disks were there for copying. The computer room was open from 7pm-5am Fri & Sat night, and was full until around 3am Saturday morning. And yes, people did get kicked out at 5am when they closed the room.
* Translator.library is being rewritten for 1.4.
* Workbench is coming out of the ROMs for 1.4.
* FFS (the Fast File System) will be in ROM in 1.4.
* Jim Goodnow of Manx, true to form, was playing with his compiler source (version 4.0) around 3am Saturday morning. He's got a flickerfixer too.
* Leo suggested CBM throw away Exec/DOS and start over.
* SubLogic demanded CBM recode everything in assembler.

...and many, many more things. There were just loads of things going on, all at once. Everyone will have their own recollections and things that stand out to them. I think a good time was had by all!

From: Marco Papa, Felsina Software, Los Angeles, CA

Bob's recollection is pretty much close to reality, though I don't know whether what I am saying makes any sense since, as everybody else, I had 12 hours of sleep in 3 days. The fun was great (especially at

the USENET meeting and Randy's taped replay of Star Trek). Dave Haynie won the USENET Boing! award, but George Robbins was VERY close second. The conference cost less than last time ($200 instead of $450) and general agreement is that we (developers) come back with more: more docs (nicely printed), more source code examples.

One thing Bob did not mention are the Janus disks that RJ handed out that allow using the bridgecard to create a 976-type partyline :-) Dr. Henry Rubin, Commodore Chief Operating Officer, seemed to enjoy RJ's talk quite a lot.

There was talk of another devcon much sooner than 1 and 1/2 years from now. The IFF library meeting also got started. I don't know if it will become one of the official Working Groups, but it is likely. Carolyn is handling that one. The Addison-Wesley Manuals are going through a major revision by Carolyn and Nancy Rains. Only drawback, I came back with a cold. It is nice to be back in "sunny" Southern California :-) How about having the next devcon in LA in January of 1989 (it was 85 degrees this year)?

Lauren, Patsy and Nancy took care of all our needs, and Gail Wellington run the show very nicely and smoothly. Is she running for a higher position inside CBM? Lots of developers were hoping so.

===

A Simple Paint Program
by Donald Welsh

Everyone loves a pretty picture, and what better way to make one than on the Amiga using a free paint program? On Fish disk #128 is a program titled, modestly, "Simple Paint" or "Paint". Paint is written in web, a language which makes 68000 assembly code on the Amiga look more like C or Amiga Basic. This makes it a very small program, under nine and a half kilobytes.

I bought Fish Disk #128 for "sed" and the disassembler. When I first investigated "paint", I dismissed it as no more powerful than programs I had seen a couple of years ago written in ABasiC. When I looked at the source file (not visible from Workbench), I saw it was more. For starters, it is the only PD paint program I've seen which uses HAM (hold-and-modify mode). HAM is that mode your mother warned you about, the mysterious one that can display 4096 colors on your Amiga monitor *all at once*. As the source is over two thousand lines long and (contrary to what the POSTER3 file says) there is no document included, I'll tell you what I found.

First, though, a warning: there is no way to save your work to disk. When you start paint, a message appears telling you to "<TYPE ESCAPE TO EXIT>". This is the way out of the program; there is no close gadget. There is a row of color blocks at the top of the screen; this is your palette. Select a color to draw with it. Select the blank space to the left of the leftmost color to use the cycle draw feature again.

The cursor is in the shape of a box. Pressing the left button deposits a square of that size of a solid color. By moving the mouse around and clicking the left button, you cycle through the available colors. You can paint a picture about four times as big as the screen. The cursor keys move the canvas around for you. There are 64 color gadgets at the top of the canvas, including one at each end which is "canvas gray". The one on the left is the "cycle draw" gadget, and the one on the right is the "canvas gray" color selector. There are no menus in paint; clicking the right mouse button changes the tool in use. The basic tools (which should be self-explanatory) are, in order:

square
horizontal bars
vertical bars
brush
line

There are more tools and more ways of using these tools, available via the modifier keys. There are four types of modifiers in paint, the left shift key, the right shift key, the alt key, and the space bar. There are two ways of applying each modifier key, both when changing tools and when drawing, and they may be used in combination. Whew! And it calls itself simple!

First of all, holding down a modifier key while changing tools allows you to use different sets of tools. In the alt-tools, the bar patterns are finer, the brush makes smaller dots, and the line tool draws dotted lines. The alt-tools square picks up whatever is on the screen under it, and deposits it wherever the left button is clicked. The left-shifted-tools paint various patterns. The first three tools are oval brushes, one to draw in a solid color, two for the bar patterns. The right-shifted-"box"-tools look just like the normal box tools, but act as brushes: three square brushes painting solid color, horizontal and vertical stripes.

The shifted-tools brush and line tools draw using patterns determined by the color in use. The space-tools affect large areas of the canvas. They are used by indicating a rectangular area by dragging the mouse from one corner to the other. The box fills the rectangle with a solid color and the bars fill it with solid color bars. The space-brush draws an open ellipse, and the space-line draws an open box. Hold down the space bar while defining the ellipse to draw a filled ellipse.

You can use these modifiers in combination. The combinations are usually obvious: alt-shift for oval brushes and fine bars, alt-space for blocks and fine bars. Shift-space gives you a big oval brush and alt-shift-space will draw fine bars with the same brush. The alt-space brush draws an open ellipse, and the line draws a dotted-outline box. The others tools allow different patterns to be used, again based on the color selected. The alt-rightshift-solidbox picks up the area under it and paints with it. The leftshift-space tools are large oval brushes.

The modifier keys may be held down while drawing. Alt generally uses the tool you are using to erase an area, or paint it to the canvas color. Thus, the easiest way to get an "eraser tool" is hold down the alt key while painting with the right-shift-tools solid block. All the bar tools and dotted line tools use background colors when the right shift key is held down while they draw. To select a background color, select a color while holding down the right shift key. The "s" and "r" keys allow you to save and restore a rectangle from the canvas to a file in RAM named "pic". There may be a trick to this, as it doesn't work consistently for me.

The small size of "Simple Paint" and the use of HAM make it an excellent demonstration of the power of the Amiga. Its greatest flaw is that it does not allow you to save your masterpieces for posterity.

Amy and Meg : The Megabyte Saga Continues

by Mark Kelly, Swan Hill

My beloved Megaboard2 died one day. Meg, the sexy little boxful of silicon I had sponsored to leave the USA and come to live with me in Australia died in her sleep. Those of you whose memories aren't dead may recall the article in last August's WORKBENCH about my purchase of Meg from America. One day I fired up Amy, my Amiga 1000, and, as usual, she rolled over and asked Meg to wake up. Meg didn't answer. Amy was distraught. Amy's screen went pale and, probably out of sympathy, she refused to boot. I desperately applied the electronic equivalent of Cardio-Pulminary Resuscitation: I pressed Ctrl-Amiga-Amiga fifteen times a minute and blasphemed abominably as Amy and Meg lay dead together in a high-tech parody of Romeo and Juliet.

With tears in my eyes and inventive invectives in my throat, I shut down Amy and prised Meg from her side. Amy immediately recovered and worked normally. `It's those [expletive deleted] PAL chips again,' I muttered. Last year Amy underwent silicon surgery at High Tech Hospital in Brighton so she and Meg could mate successfully. It looked like a relapse. I packed Amy and Meg back into my Starion and returned to High Tech. The technician exercised his considerable skill by plugging Meg into another Amiga and reporting that it was dead. Amy was fine. He charged me $20 for this arduous work. He must've done his training with some medical specialists I know...

`The repair is under warranty,' the young shop assistant said amiably as he took my money. `If you have any trouble...' `What repair?' I muttered. I've learnt to mutter a lot lately. `Ah...' he muttered back, biting his lip.

I had two choices. Place Meg in the middle of the Nepean Highway and wait, or send her on an all-expenses-paid holiday back to America. After digging out Meg's long-expired warranty card I found I needed a `Return Merchandise Number' before I returned her. I dutifully rang Progressive Peripherals and Software, Meg's parents, and they gave me a number. I STILL don't know what it was for. I packed Meg in a cosy box with a couple of sandwiches for the trip and posted her. I really didn't expect to ever see her again. That was on April 8th. Life without Meg was a trial for Amy and me. I had to live without the joys of my crash-proof VDO: and I felt suffocated.

Four weeks later I nearly had heart seizure. MEG WAS BACK!! PP&S has fixed her beautifully and had charged nothing whatsoever - not even for return postage and insurance which cost $US19! I was truly astonished not only at the speed of their work but at their benevolence: Meg was well and truly out of warranty! As a comparison, I sent an order for an Amigan Disk off to a certain users group in Boronia shortly after posting Meg to America. Two weeks after Meg's return, I'm still waiting for the PD Disk. Still, Meg and Amy are together again and they're happy. So am I, thanks to PP&S.

Since I have finished paying my car off and the finance company has released my mother from custody, I have had been lashing out somewhat. I ordered AC-BASIC, an AmigaBasic-compatible compiler, from GO-AMIGO. Six weeks later I was turbo-boosting my old BASIC programs and I'm very pleased with the results. I've never managed to get into deep C programming and being able to compile my BASIC code has proved to be a satisfactory substitute. I can recommend AC-BASIC for serious BASIC programmers (you C boffins out there can stop sniggering. There IS such a thing as serious BASIC programming!) The executable code is admittedly bulky but that's the price you pay for programming convenience, I suppose.

The negative side of the GO-AMIGO order was that I had also asked them to send FACC2, the intelligent disk-caching utility to accelerate floppy disk access. They didn't send it but they did charge me for it. I immediately wrote to GO-AMIGO asking politely for satisfaction. Inwardly I knew that I was a victim of another mail order house and that even The Investigators wouldn't be able to help me. I was in for another jolt. Much as I deride the Americans down for chauvinism and their massacre of the English language, I am constantly surprised by their business acumen. They, unlike most Australian outfits, have realised that happy customers are repeat customers. I soon received from them a generous personal apology for their error and a promise that FACC2 was winging its way to me.

In my never-ending search for the ultimate word processor, I eagerly sent off for Amigan #14 (as mentioned above) which contains WordWright, an interesting-sounding free word-churner. Unfortunately my patience withered after the six-week wait and I bit the proverbial bullet and sent off for WordPerfect. I had recently priced it in Melbourne and knew that a local purchase would mean the finance company would be holding my mother forever as security. I know overseas purchases can cause trouble with customer support, but comparing $US200 with $OZ600 is a rip off in anyone's language. Anyway, after experiencing the `support' offered from local distributors, I feel I wouldn't be missing anything. I also threw caution and my VISA card statements to the wind and ordered a TIMESAVER, a battery-backed clock and keyboard macro gizmo that also acts like CONMAN. When it arrives, I'll let you know how it behaves. I think I'll call it Tim. Amy and Meg would love another playmate.

BASIC Pictures

by Tony Giles

I remember when I first got my Amiga, envisaging all these wonderful programs I would write. Having some previous experience with BASIC, I dived right in.

One of the first things I had planned was a simple BASIC program that used a picture drawn in Deluxe Paint II. So I opened up my AmigaBasic Manual, expecting it to tell me all about it. No need to tell you what I found in there. So, not being one to be detered so easily, I set about trying to find a good book that told me how to do everything possible on the Amiga.

I'm still looking.

I finally resorted to deciphering general purpose programs for loading pictures into AmigaBasic, and trying to understand them enough, to write a short routine for loading a particular picture.

After a lot of frustration, late nights reading, and endless trial and error. I finally found that the best way (so far) was to convert the IFF picture to ACBM format (using the LoadILBM-SaveACBM program on Extras 1.2) and use a GREATLY reduced version of the LoadACBM program by Carolyn Scheppner. This method has its pro's and con's. The picture file, once in ACBM format takes up more memory (approx 51386 bytes for a 32 color 320*256 picture) than in IFF format. The advantages are, that its easier to load and MUCH faster than any other method I've seen.

The first thing you need to do is to convert your picture to ACBM using the program in the BasicDemos drawer of your Extras disk. If your picture was done on a PAL version of deluxe paint (or any other PAL graphics program), when you run LoadILBM-SaveACBM, it will draw your picture in interlace. Don't worry, the program was written with NTCS monitors in mind (Americans!!!!). You can still load the ACBM picture on a normal 256 pixel height screen.

Next, you need to copy three .BMAP files from the BasicDemos drawer on the Extras disk to the same directory that your basic program is in. These .BMAP files are:

Dos.BMAP
Exec.BMAP
Graphics.BMAP

These files allow you to access the Dos, Exec & Graphics libraries from your Basic program. This gives you use of the functions listed in the ROM Kernal Manual (Any one stupid enough, to try and read it will know what a vicious piece of literature it is), which can be quite useful for doing all sorts of things that AmigaBasic would normally not be able to do.

Before you actually start programming, you need to get some information about the picture to be loaded. The picture loading is made a lot easier if the color palette of the picture can be defined without actually reading the values from the ACBM file. To do this, we need to obtain the RGB values of each color in the palette. If you load the IFF version of your picture into DPaint II, you can get these values.

Looking at the color menu on the bottom left of the DPaint II screen. The top left color block is color 0. The color directly below is color 1, below that is color 2 , and so on, until you reach the bottom block of color. The next color in the sequence is the top block of the next column (the one to the right of color 0), it then counts down to the bottom of that column and so on. Below is an example for 32 color registers :-

0	8	16	24
1	9	17	25
2	10	18	26
3	11	19	27
4	12	20	28
5	13	21	29
6	14	22	30
7	15	23	31

Now, if you open the window for changing the colors, you will see three vertical sliders with R, G & B above them. Down the left side of the sliders is a number scale from 0 to 15. Using the pointers in the sliders and the number scale, it is possible to get an R (RED) value, a G (GREEN) value, and a B (BLUE) value for each color. Write down the RGB values, in order, for each of the colors in your palette, noting the color register No, beside each set of values. Next you need to perform a small calculation on each set of RGB values. The calculation is

```
TOTAL  =  ((256*Red)+(16*Green)+Blue)
```

Where TOTAL is that color's value. This should result in a number between 0 & 4095. 0 signifies black, and 4095 is white. Write down each color register's final value.

While we're still in DPaint II, just make sure you have the correct dimensions of the picture. Eg. 320*200 or 640*256 etc.

Now the last thing we need to know is the size, in bytes, of the ACBM file containing your picture. This can be found by doing a LIST of the directory it is contained in.

OK, Now we can do some programming.

First thing to do is DIM a variable bPlane&(x), where x is the number of bitplanes in your picture. The number of bitplanes is relational to the number of colors in the picture. eg.

No of colors	Bitplanes
2	1
4	2
8	3
16	4
32	5

Therefore, if your picture uses 16 colors, the DIM statement would look like :-

```
DIM bPlane&(4)
```

Next, we have to declare some functions that will give us access to the Dos, Exec & Graphics libraries:

```
DECLARE FUNCTION xOpen&  LIBRARY
DECLARE FUNCTION xRead&  LIBRARY
DECLARE FUNCTION xWrite& LIBRARY
DECLARE FUNCTION AllocMem&() LIBRARY

LIBRARY "dos.library"
LIBRARY "exec.library"
LIBRARY "graphics.library"
```

The function called AllocMem&() will be used to allocate a private block of memory where we will load our color table, and also do a dummy read of our picture file.

The rest of the code to load the picture can all exist in a sub-routine such as LoadPic. So when you wish to load the picture, you just use:

```
GOSUB LoadPic
```

For the sake of this explanation, I will open the screen and window for the picture inside this sub-routine. There is no reason why you can't open the screen & window in your main program, or another sub-routine, and then call LoadPic when you wish to put the picture in that window.

Wherever you open your screen & window, you need to find the memory address of the window, and then use this address to find the address of the screen. This is done as follows :-

```
SCREEN 2, 320, 256, 5, 1      'opens a screen 320*256
                              'with 32 colors.
WINDOW 2, "pic", , 0, 2       'opens a plain window
                              'same size as screen.
sWindow& = WINDOW(7)          'loads the address of
                              'window into sWindow&
sScreen& = PEEKL(sWindow& + 46) 'loads address of
                                'the screen into
                                'sScreen&
sBitMap& = PEEKL(sScreen& + 88) 'loads address of
                                'screen bitmap into
                                'sBitMap&
```

Using the sBitMap& value, it is possible to find the memory address for each of the screen's bit-planes. The variable bPlane&() that was dimensioned above will hold these addresses.

```
FOR x = 0 to (Number_of_planes-1)
   bPlane&(x) = PEEKL(sBitMap& + 8 + (x*4))
NEXT x
```

Now, we need to set up our color table, but before we can do this, we must allocate a block of memory to load it into. This is done as follows :-

```
mybuff& = AllocMem&(300&, 65537&)
```

The 300& is the size (in bytes) of the memory block we have requested. The 65537& specifies the type of memory we want. The number is obtained as follows :-

1	Stable, non-relocatable memory area
2	Chip memory, the lower 512k
4	Fast memory, memory above 512k
65536	Memory area is cleared

To select one or more of the four options, you just add their numbers together to get a total. We chose 65537, which is Stable, non-relocatable memory that is cleared for us.

The variable mybuff& used above, has the beginning memory address of our block of memory loaded into it. Thus, when we want to access our memory, we use mybuff& as a pointer to it.

Now, we have our memory area and a pointer to it, we can load our color table into it. But first, before we do, it would be a good idea to black out the screen, so that the picture loading can't be seen. To do this, we call a library function that sets the values in the color registers.

```
CALL LoadRBG4((sScreen& + 44), mybuff&, No_of_Colors)
```

The (sScreen& + 44) is a pointer to the Viewport for the screen. mybuff& is the beginning address of our memory. This is where the values to be placed into the screen's color table, are read from. Since our memory area is clear (full of zeros), all the colors are set to zero (or BLACK).

No_of_Colors is the number of colors in your picture. If you also want to black out the mouse pointer, just add 4 to this value.

Now we can load our color table into our memory. Each color value will take up 2 bytes of memory in the block we allocated. Thus 16 colors would take 32 bytes to store. To load the values into our memory, you POKEW (poke a word, or 2 bytes) each color value in succession, starting at the base address of our memory. eg:

```
POKEW mybuff&, Color_0_Value
POKEW (mybuff& + 2), Color_1_Value
POKEW (mybuff& + 4), Color_2_Value
etc
```

Where Color_0_Value is the value for color 0 etc. Once all the color values are loaded, we can open the picture file for loading. This is done as follows :-

```
picture$ = "name-of-picture-file" + CHR$(0)
fhandle& = xOpen&(SADD(picture$), 1005)
```

The SADD() function returns the address of the first byte in a string. Therefore, SADD(picture$) will pass a pointer to the name of the picture file, to the library function xOpen&().The value 1005 tells the xOpen&() function that the file to be opened is an existing file. The xOpen&() function opens the file and loads a pointer to it into the variable fhandle&. So, from now on, when we wish to access our picture file, we use fhandle& as a pointer to it.

Now, we're ready to load the picture, but first I need to explain a bit about the way the ACBM file is layed out. At the start of the file is all the information about what type of picture it is. The dimensions, colormap and other info that we don't really need to know.

The last part of the file is made up of the bit planes of the picture. This is the only information we actually need from the file.

So, the problem is, finding out how much rubbish is at the start of the ACBM file. To do this, we need to make another calculation. Knowing all the dimensions of the picture, it is possible to calculate the size (in bytes), that each bitplane will take up in the file. Once we know this, it is easy to work out how much rubbish is at the start of the file

The size of each plane is equal to the picture width in pixels, multiplied by the height, and then divided by eight. We then multiply this result with the number of bitplanes to find the total size of the full bitmap. Subtracting this amount from the file size (in bytes, which you looked up before using LIST), will give us the amount (in bytes) of rubbish to ignore when we read the file.

Therefore, the calculation would look like this :-

```
BitPlaneSize = (Width * Height) / 8
BitMapSize = BitPlaneSize * NumberOfBitPlanes
Rubbish = FileSize - BitMapSize
```

For a 320*256, 5 plane picture whose ACBM file is 51,386 bytes , the calculation would be

```
(320*256)/8 = 10240
10240 * 5 = 51200
51386 - 51200 = 186 bytes of rubbish.
```

Now, to set the pointer for the xRead&() function to the start of the first bitplane, we need to do a dummy read of 186 bytes.

```
rLen& = xRead&(fhandle&, (mybuff& + 100), 186)
```

fhandle& is the pointer to our picture file. (mybuff&+100) is a pointer to our block of memory, but 100 bytes into it, so that we don't write over our colormap. And the 186, is the amount of bytes to be read from memory.

Now, we can load our picture.

```
FOR x = 0 TO (No_of_planes - 1)
  rLen& = xRead&(fhandle&, bPlane&(x), Size_Of_BitPlane)
NEXT x
```

That's it, our picture is now on the screen. Only, we can't see it because all the colors are set to black.

```
CALL LoadRGB4&((sScreen& + 44), mybuff&, No_of_colors)
```

and there is our picture.

So, a working program, using a subroutine to load the picture, would like the example below :-

```
REM This program loads a 32 color, 320*256 picture.

DIM bPlane&(x)

DECLARE FUNCTION xOpen&  LIBRARY
DECLARE FUNCTION xRead&  LIBRARY
DECLARE FUNCTION xWrite& LIBRARY
DECLARE FUNCTION AllocMem&() LIBRARY

LIBRARY "dos.library"
LIBRARY "exec.library"
LIBRARY "graphics.library"

Main:
   GOSUB LoadPic
   FOR count = 1 TO 10000:NEXT count
Finish:
   WINDOW CLOSE 2
   SCREEN CLOSE 2
END

LoadPic:
   fhandle& = 0
   mybuff& = 0
   SCREEN 2, 320, 256, 5, 1
   WINDOW 2, "pic", , 0, 2
   sWindow& = WINDOW(7)
   sScreen& = PEEKL(sWindow& + 46)
   sBitMap& = PEEKL(sScreen& + 88)
   FOR x = 0 TO 4
      bPlane&(x) = PEEKL(sBitMap& + 8 + (x*4))
   NEXT x
   mybuff& = AllocMem&(300&, 65537&)
   IF mybuff& = 0 THEN
      PRINT "couldn't allocate memory"
      GOTO Finish
   END IF
   CALL LoadRGB4((sScreen& + 44), mybuff&, 32)
   FOR x = 0 to 62 STEP 2
      READ value
      POKEW (mybuff& +x), value
   NEXT x
   picture$ = "picture-name" + CHR$(0)
   fhandle& = xOpen&(SADD(picture$), 1005)
   IF fhandle& = 0 THEN
      PRINT "couldn't open picture file"
      GOTO Finish
   END IF
   rLen& = xRead&(fhandle&, (mybuff& + 100), 186)
   FOR x = 0 TO 4
      rLen& = xRead&(fhandle&, bPlane&(x), 10240)
   NEXT x
   CALL LoadRGB4&((sScreen& + 44), mybuff&, 32)
   CALL xClose&(fhandle&)      'closes file
   CALL FreeMem&(mybuff&, 300&) 'de-allocs mem
RETURN

DATA  col0, col1, col2, col3, col4, col5, col6, col7
DATA  col8, col9,col10,col11,col12,col13,col14,col15
DATA col16,col17,col18,col19,col20,col21,col22,col23
DATA col24,col25,col26,col27,col28,col29,col30,col31
```

Hopefully, this will give some insight into using pictures with AmigaBasic. If your prepared to do some research into other functions contained in the ROM Kernal manuals, it is possible to do many things with screens etc. A lot of functions are accessible, once you know how to use the libraries.

==

A Look at Jet

Background

When I bought my first computer (an eight bit Atari) I thought the most wonderful piece of software out was Sublogic's Flight Simulator. It had one drawback however. It was boring. This was not the fault of the program since it accurately reflected the boredom of puttering along at about 100 knots when all reference points (ie the ground) were 5000 feet below.

There was one area of the product which had the potential to be an interesting game. This was the World War Ace section. In this you piloted an old biplane through the sky and did battle with a number of similar aircraft. Alas this potential was not fulfilled due to some short comings in the game.

Firstly, although you plane could only shoot forward the enemy guns could shoot in any direction. Thus if by clever manouvering you got behind an enemy, you gained no advantage. Secondly the enemy seemed to defy the laws of physics by flying sideways and even backwards. Thirdly because they also had radar, you could never catch one unawares and get behind. Thus all dogfights were head on charges.

Since you were out-numbered it reduced the game to the level of the old arcade games where you attempted to shoot moving pictures of planes which were always facing you. Flight Simulator on the Amiga was a huge step forward graphically but the game play was no better. Thus there was still not a dogfight simulation on the market. When I first heard of Jet I was hopefull that it would fill this gap. Now it has arrived I must review it with mixed feelings.

TheProduct

Jet puts you in the cockpit of a modern jet fighter bomber. You select various armaments and go on a mission to either bomb or dogfight. The scenery depends on the mission selected. Some of it is rather basic, possibly to speed up response while some is highly detailed and quite entertaining just to fly over.

Jet is in many ways similar to Flight Simulator. Both use the same graphics techniques although in addition Jet has the painted backdrop which remains at the horizon. To speed the screen refresh rate this can be switched off. Both use the same number of colours. There had been a rumour that to improve screen update rates Jet would only have four colours but this has proved false. Both have the same screen update rate, but Jet can seem a little slower because of the greater movement possible between two screen shots. This is caused by the higher speeds a jet travels at. Both have a number of views for the pilot and both can use the same scenery disks. Actually the scenery on the Jet disk is a bit limited and you can't dogfight over imported scenery.

The Gameplay

The jet uses a "fly by wire" system of control. This allows the pilot to point the jet and an on-board computer will make all the adjustments needed to maintain that attitude of flight. Thus it is much easier to fly than Flight Simulator.

However in the dogfight there is a problem. The problem is that a modern jet flies well above the speed of sound. If two jets detect each other on radar at 25 Km they could meet in about one minute. By using air to air missiles the battle would be over before the actually got close enough to see each other properly. While this may be the way real modern warfare happens, it does reduce the feeling of being involved.

To get around this the manufacturers made a few choices. Firstly they gave you the ability to survive multiple missile hits. This allows you to get in close enough to use guns. Then to even the odds up they always sent you out against a number of enemy. Thus you would fire an initial salvo of missiles, and by the time you knew if they had hit their targets you were getting hit by the enemy missiles (which despite the documentation, are almost impossible to avoid). By the time you recovered from this you were in the dogfight proper and could loop about trying to shoot them down. Perhaps a better solution would be to remove the radar and thus give both sides the ability to sneek up on the other undetected.

It appears that the speed of the aircraft involved are not realistically rendered. If a plane 200 metres away flies across your path at mach 1 you would expect it to be out of sight in less than a second. Yet it takes several seconds to pass. Despite this concession to game play the speed makes it very difficult to follow another jet in a dogfight. Basically jets are just too fast and manouverable. One saving grace here is the ability to to set the screen to wide angle which makes it easier to follow the path of an enemy in close. There is a further problem with the dogfights. Although the manual states the tracking system will lock on to an enemy once he is in range of your weapon this seems limited to a maximum of about 5 miles. This is the range of the AIM-9 missiles it is only a fraction of the AIM-7 missiles' 25 mile range. It is unclear whether this is a bug or a limitation of the targetting system.

The Bugs

From time to time your jet will explode for no reason. This happens on about 10% of flights. The manual states you will not be chased in the safe air surrounding your airbase. This is not true. They will even fire missiles at you while you're on the ground. The ground targetting system identifies SAM sites but not primary targets. Thus these can only be detected visually.

In combined (dogfight & bombing) mode the targetting (but not highlighting) of ground targets still works. You fire an air to ground missile and watch as it tracks to a primary target which was not the actual object you were aiming at. In bombing only mode it is even worse since not only are the targets not highlighted but the system identifies targets at incorrect locations. Thus when you line up with a destroyer and drop a bomb the bomb does a 90 degree turn and then moves off to the distance seeking some target.

Bombs (which have no propulsion and thus are limited in range) sometimes act like missiles and travel 50km or more past the target. The range circle is ment to to inform the pilot if the selected ground target is in range of the selected weapon. Unfortunately it shows only the range for bombs, and not air to ground missiles.

The Conclusion

The game has a lot of potential. Stores inform me that Sub Logic are aware of the bugs, and hopefully they will release a fix soon. In the meantime, the game is crippled.

The Signals at Boot

[Editor's note: This article was originally published in the March/April 1988 issue of The Amigan Apprentice and Journeyman, and is Copyright 1988 by the Amigans, PO Box 411, Hateras, NC 27943, USA]

There's rhyme and reason to the colours on Amy's screen when you cold boot. Dark grey means the microprocessor is running okay; light grey, that the ROMs passed a checksum test; white, that RAM is tested and system startup proceeds. That's the normal course of a boot.

But, if something is wrong, you should see red for a ROM error, green for an error in chip RAM, blue for an error in the custom chips, or yellow if an error occurs before the error-trapping routines (which give you the Guru) are up and running.

Even the keyboard has a self-test routine; if that fails, the caps lock light light blinks. One blink means that a check of keyboard ROM failed; two, that keyboard ram is not good; three, that the keyboard's internal timer isn't scanning the keyboard properly; four, a short circuit in the keyboard. If a line of apostrophes forms at boot, your keyboard connections are fouled up.

Such diagnostics can tell you (and the folks at the repair shop) where any problems lie. For more details, see The Transactor for March, '88, p. 74.

The PopCLI Problem

By Bill Miles

Here is the answer to Peter Kinross' question about PopCLI in last months Workbench.

The problem is that PopCLI accesses the disk when you press the 'Hot Keys' to create (evolve!) a new CLI. The cause is that PopCLI uses the Execute() function from the AmigaDOS library, to execute the NEWCLI command. The Execute() function in turn uses the RUN command to execute the NEWCLI command (that's clear isn't it ?!). As stated in the AmigaDOS Developers Manual, for the Execute() function to work, the RUN command must be in the C: directory. This is why the disk is accessed, to load C:RUN. I have come up with two ways around this problem.

Solution 1

In your startup-sequence assign C: to your RAM disk command directory. This is the best solution if you copy the whole C: directory to RAM. The commands RUN and NEWCLI must be in this directory. You may want to add SYS:C to the command search path so that the CLI can still find commands on disk. eg.

```
ASSIGN C: RAM:C
PATH SYS:C ADD
```

Solution 2

This patch is only suggested to those who understand what they are doing. If you don't want to change the C: assignment, you can fix the problem by editing the Kickstart disk. Just change the C:RUN used by the Execute() function into R:RUN. Then in the startup-sequence assign R: to your RAM disk command directory, the RUN and NEWCLI commands must be in this directory. You will also need to specify the full RAM disk pathname to NEWCLI in the PopCLI arguments. Otherwise when you press the 'Hot Keys' RUN will get the NEWCLI command from disk. This is because as a new task C: and DF0: are the only paths available initially. Remember PATHs are local to each task whereas ASSIGNments are global to all tasks. eg.

```
ASSIGN R: RAM:C
POPCLI 300 RAM:C/NEWCLI > NIL: FROM RAM:S/NEWCLI.S-S
```

Note the FROM file in the POPCLI arguments, this is a sort of mini startup-sequence for NEWCLI's. This is a handy idea, at present mine only adds the ram disk command directory path. eg.

```
PATH RAM:C ADD
```

The C:RUN string can be found on the Kickstart disk Block 447 (decimal), that's Track 20, Sector 7, Side 0. You can use Sectorama (Fish 108) to change the 'C' to 'R'. The Kickstart checksum must be corrected, possibly by using Sumkick (Amigan 12).

Missing Macros In The Assembler "include" Files

by Allan Duncan

In order to maintain a symbolic link between the programmer and the machine language requirement for numeric constants, Commodore has provided a number of files to program developers, and anyone else who cares to lay down their money. In AmigaBASIC there are the BMAP files which have been generated from the FD files. Both these give the offset into a library of the specific routine asked for. In assembler there are a couple that I was interested in as they are in the startup code for ARP V1.1, namely "exec/exec_lib.i" and "libraries/dos_lib.i". In the interests of confirming that I could regenerate the code segments that were provided, I set to with the Manx C compiler and assembler, only to produce heaps of error messages. Tracing the source of the error is not helped by the lovely quirk of Manx to give you a listing file with just the heading when there is a problem, just the time when the offending code is most likely to be looked at.

Probing into the code on the principle that if they can, I can, I tracked the source of the difficulty to this line -

```
        INCLUDE    "exec/exec_lib.i"
```

The assembler barfs at it because the include has a reference to a macro FUNCDEF. Just in case there was something not in the Manx assembler, I tried the original Metacomco one, to no avail, however on the disk I found a file that contained the macro, which referred to yet another include file!

So, to get any of these files that mention exec/exec_lib.i to assemble you need to precede it with .....

(a) include "exec/libraries.i" which contains this

```
*------ Special Constants ---------------------------

LIB_VECTSIZE    EQU  6
LIB_RESERVED    EQU  4
LIB_BASE        EQU  $FFFFFFFA * (-LIB_VECTSIZE)
LIB_USERDEF     EQU  LIB_BASE - (LIB_RESERVED * LIB_VECTSIZE) music
LIB_NONSTD      EQU  LIB_USERDEF
```

and .....

(b) the missing macro, from "ovr.asm" on the ASSEM-DEVEL disk of way-back-when -

```
* Kludge to determine the correct LVO XREF's:

FUNCDEF   MACRO      * function
_LVO\1    EQU        FUNC_CNT
FUNC_CNT  SET        FUNC_CNT - 6
          ENDM

FUNC_CNT  SET        LIB_NONSTD
```

The effect of this include is to generate the numeric offsets and relate them to the symbolic names we all know and love. This, you would expect, would be the same for all of the libraries, but no, for inconsistent behaviour by Commodore, if you have ....

```
          INCLUDE    "libraries/dos_lib.i"
```

THIS starts with :-

```
* Library interface offsets for DOS library

reserve   EQU  4
vsize     EQU  6

count     SET  -vsize * (reserve + 1)

LIBENT    MACRO
_LVO\1    EQU  count
count     SET  count - vsize
          ENDM
```

which passes with flying colours, without having to hunt around for the macro.

It is interesting to note that both libraries start the offsets at -30, despite the LIB_NONSTD used in FUNCDEF. Maybe the NONSTD is a carryover from the early exec.library shown in the RKM that started at 0. It is for reasons like this that the "exec_lib.i" and the ability of the OpenLibrary call to specify a version number exists.

==

Australian Public Domain Software Releases

Drac (Bohdan Ferens), sysop of our AmigaLink BBS, has started an Australian public domain collection, and below he lists the contents of the first two disks.

AUG PD #1

DX7 — A voice editor for people with MIDI interfaces.
Author: A.V. Ermin (Ermin & Associates)

Game — A game based on the good old "shoot them up" approach. It makes extensive use of bobs and vsprites.
Author: Paul Vella

AUG PD #2

This a collection of Delux Music Scores from a member of the Amiga Users Group in Melbourne. Please read the text files as they contain information on which instruments are needed to play these scores. The titles included here are:-

Diamonds on the soles of her shoes
Graceland
The Boy In The Bubble
You Can Call Me Al
Eleanor Rigby
You're The Voice
Bolero
Allegro Fuer Elise
Sonata Facile
Birthday
gypsy
Music Box
Homeless
Under African Skies
O Come All Ye Faithful
Thornbirds
Bereite dich, Zion
Brandenburg No.2
Piano Concerto No.1
Triumphal March
Day Tripper
Lucy

PatternMaster — PatternMaster is a versatile pattern editor enabling the programmer to design fill patterns for the Amiga. These patterns can then be saved by using one of four options:

1) Save as Pattern file
2) Save as IFF Brush
3) Save as Bob
4) Save as Basic Data

Author Alan Garner

DUX — Directory Utility DUX version 6.0. This version is a major tidy-up and enhancement of the DUX5 program on Fish disk 67. In general the aim has been to make the program easier to use, and to provide a wider range of functions. I have plenty of ideas for further improvements, but it is hard to find the time! So I offer this as it is, and will post future enhanced versions as they are produced.
Enhancements by Alan Fieldus

SayWords — Here is a silly programme. It all began on the Commodore 64, I wrote a programme to produce silly words using the RND and MID$ statements so that if I ever got around to writing an adventure (I never did) I would never be stuck for unusual names for things such as places, characters or objects. Of course I then adapted it for the Amiga using the SAY and TRANSLATE$ commands to actually speak the words. These programmes were written entirely for amusement and as such are quick and dirty, none of them contain a great deal of error trapping and should be used with care.
Author: John Elston

==

Newsletter Back Issue Order Form

Issue Numbers:

Number of issues at $2 each $

Less 10% for 5 or more - $

Club Use Only Total $

Mail to: Amiga Users Group, PO Box 48, Boronia, 3155

Member's Name:

Address:

SOFTWARE ORDER FORM

Disk numbers :

Disks supplied by Amiga User Group @ $8 $

Disks supplied by member @ $2 $

Club Use Only
Receipt #: Mailed on: / / Total $

Mail to: Amiga Users Group, PO Box 48, Boronia, 3155, Victoria.

Member's Name:

Address:

Application for membership of The Amiga Users Group Inc

Membership is $20 per year. Send your cheque to: **Amiga Users Group Inc, PO Box 48, Boronia, 3155**

Surname: ______________________

First name: ______________ (no initials)

Address: ______________________

______________ Postcode: ______

Phone Number: __________ STD Code: ______

What services would you like AUG to provide: ______

____

____

Signed: ______________ Date: ______

Details on this side are optional

Year of birth: ______ Which model Amiga: ______

Occupation: ______________________

Interests: ______________________

____

____

Dealer's Name: ______________________

Dealer's Address: ______________________

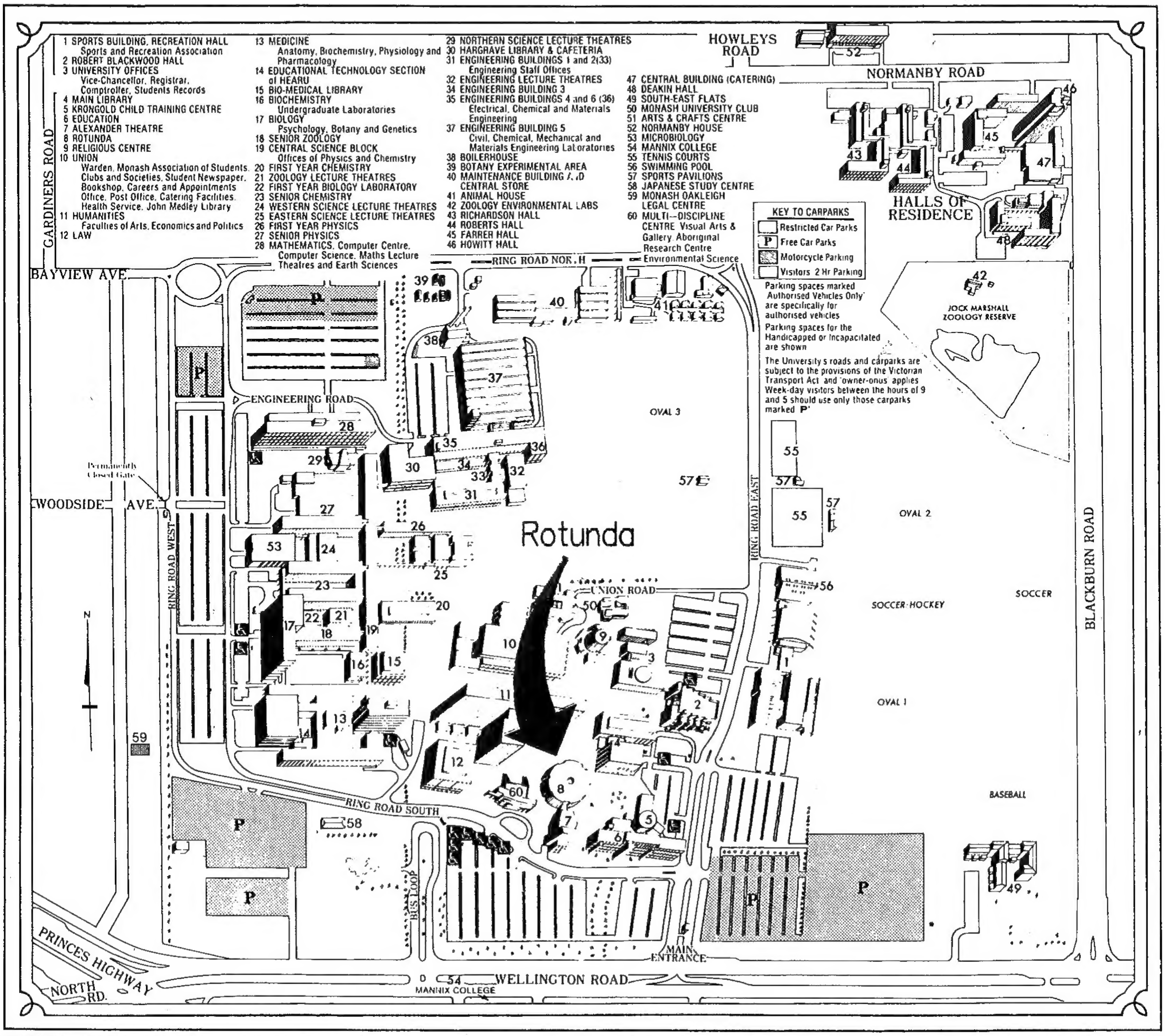
______________ Postcode: ______

In the event of my admission as a member, I agree to abide by the rules of the Association for the time being in force.

Club Use Only	Date	Paid	Rcpt #	Memb #	Card Sent

AUG now meets on the third Sunday of each month

Monash University is in Wellington Road, Clayton. See Melways Map 70, reference F10. Melways map 84A shows the University Campus in details. I've drawn a huge arrow on the map below to show where the Rotunda is. The best place to park your car is the car park area between Wellington Road and the Rotunda. The entrance to the Rotunda is virtually at the point of the arrow.


1 SPORTS BUILDING, RECREATION HALL Sports and Recreation Association
2 ROBERT BLACKWOOD HALL
3 UNIVERSITY OFFICES Vice-Chancellor, Registrar, Comptroller, Students Records
4 MAIN LIBRARY
5 KRONGOLD CHILD TRAINING CENTRE
6 EDUCATION
7 ALEXANDER THEATRE
8 ROTUNDA
9 RELIGIOUS CENTRE
10 UNION Warden, Monash Association of Students, Clubs and Societies, Student Newspaper, Bookshop, Careers and Appointments Office, Post Office, Catering Facilities, Health Service, John Medley Library
11 HUMANITIES Faculties of Arts, Economics and Politics
12 LAW
13 MEDICINE Anatomy, Biochemistry, Physiology and Pharmacology
14 EDUCATIONAL TECHNOLOGY SECTION of HEARU
15 BIO-MEDICAL LIBRARY
16 BIOCHEMISTRY Undergraduate Laboratories
17 BIOLOGY Psychology, Botany and Genetics
18 SENIOR ZOOLOGY
19 CENTRAL SCIENCE BLOCK Offices of Physics and Chemistry
20 FIRST YEAR CHEMISTRY
21 ZOOLOGY LECTURE THEATRES
22 FIRST YEAR BIOLOGY LABORATORY
23 SENIOR CHEMISTRY
24 WESTERN SCIENCE LECTURE THEATRES
25 EASTERN SCIENCE LECTURE THEATRES
26 FIRST YEAR PHYSICS
27 SENIOR PHYSICS
28 MATHEMATICS, Computer Centre, Computer Science, Maths Lecture Theatres and Earth Sciences
29 NORTHERN SCIENCE LECTURE THEATRES
30 HARGRAVE LIBRARY & CAFETERIA
31 ENGINEERING BUILDINGS 1 and 2(33) Engineering Staff Offices
32 ENGINEERING LECTURE THEATRES
34 ENGINEERING BUILDING 3
35 ENGINEERING BUILDINGS 4 and 6 (36) Electrical, Chemical and Materials Engineering
37 ENGINEERING BUILDING 5 Civil, Chemical, Mechanical and Materials Engineering Laboratories
38 BOILERHOUSE
39 BOTANY EXPERIMENTAL AREA
40 MAINTENANCE BUILDING AND CENTRAL STORE
41 ANIMAL HOUSE
42 ZOOLOGY ENVIRONMENTAL LABS
43 RICHARDSON HALL
44 ROBERTS HALL
45 FARRER HALL
46 HOWITT HALL
47 CENTRAL BUILDING (CATERING)
48 DEAKIN HALL
49 SOUTH-EAST FLATS
50 MONASH UNIVERSITY CLUB
51 ARTS & CRAFTS CENTRE
52 NORMANBY HOUSE
53 MICROBIOLOGY
54 MANNIX COLLEGE
55 TENNIS COURTS
56 SWIMMING POOL
57 SPORTS PAVILIONS
58 JAPANESE STUDY CENTRE
59 MONASH OAKLEIGH LEGAL CENTRE
60 MULTI-DISCIPLINE CENTRE Visual Arts & Gallery, Aboriginal Research Centre Environmental Science
KEY TO CARPARKS
Restricted Car Parks
P Free Car Parks
Motorcycle Parking
Visitors 2 Hr Parking
Parking spaces marked 'Authorised Vehicles Only' are specifically for authorised vehicles
Parking spaces for the Handicapped or Incapacitated are shown
The University's roads and carparks are subject to the provisions of the Victorian Transport Act and 'owner-onus' applies Week-day visitors between the hours of 9 and 5 should use only those carparks marked P'
HOWLEYS ROAD
NORMANBY ROAD
HALLS OF RESIDENCE
GARDINERS ROAD
BAYVIEW AVE
RING ROAD NORTH
JOCK MARSHALL ZOOLOGY RESERVE
ENGINEERING ROAD
OVAL 3
Permanently Closed Gate
WOODSIDE AVE
RING ROAD WEST
RING ROAD EAST
Rotunda
OVAL 2
UNION ROAD
SOCCER HOCKEY
SOCCER
BLACKBURN ROAD
N
OVAL 1
RING ROAD SOUTH
BASEBALL
BUS LOOP
PRINCES HIGHWAY
NORTH RD.
MAIN ENTRANCE
WELLINGTON ROAD
MANNIX COLLEGE

BY PUBLIC TRANSPORT . . . The simplest method is to take a train from Flinders Street or Loop stations on the Dandenong/Pakenham line to either Huntingdale or Clayton. Buses run from these stations to the campus or there is a taxi rank at Clayton. With suitable connections the trip takes about 45 minutes — but it can take longer! An inner neighborhood ticket will take you all the way via Huntingdale station and the bus, but you will need to purchase a comprehensive ticket for the trip via Clayton, which encompasses two neighborhoods. The campus is also served by buses from Box Hill, Blackburn, Belgrave, Chadstone, Jells Park-Glen Waverley, Dandenong-Mulgrave, Oakleigh and Elwood.

FROM THE CITY BY CAR . . . An easy route is along St Kilda Road or Kingsway/Queens Road and then on to Dandenong Road. The campus's tall Menzies Building comes into view a kilometre or so before the left turn into Wellington Road on which the main entrance is located. Allow 40-50 minutes for the trip. Drivers should note that restrictions apply in some car parks weekdays 9 a.m. to 5 p.m. and fines **do** apply. There is ample unrestricted parking and, closer to buildings, designated two hour visitor car parks — check the map or ask at the Gatehouse.